WORKSHEET

9.5 Taking turns

Talking to each other · Robot club · about 20 min

Name _____ Class _____ Date _____

What this lesson is about

One robot at a time through a corridor: announce, wait, go.

- Listen to the rhythm
- Wait, announce, go
- Straight down the middle

Questions

6 marks in all

1. In the corridor task, which message tells you it is time to go? [1 mark]

- ☐ **A** `corridor clear`
- ☐ **B** `corridor busy`
- ☐ **C** `entering`
- ☐ **D** `out`

2. The corridor stays clear for about 9 seconds, and you need about 6 seconds to get through. After hearing `clear`, how many seconds can you afford to delay before setting off? [1 mark]

3. What does this program print? [1 mark]

```
batches = [[("Porter", "corridor busy")], [], [("Porter", "corridor clear")]]
clear = False
checks = 0
for batch in batches:
    checks += 1
    for sender, text in batch:
        if "clear" in text:
            clear = True
    if clear:
        break
print("clear" if clear else "waiting", "after", checks, "checks")
```

4. Put the steps for getting through the corridor in order. [1 mark]

Number the lines 1 to 4 to put them in the right order.

___ `Drive through, holding the middle`
___ `Wait for corridor clear`
___ `send("entering")`
___ `send("out")`

5. The Porter does not listen to messages. Why send `entering` anyway? [1 mark]

- ☐ **A** A protocol is the agreement: a robot that did listen could hold back for you
- ☐ **B** The Porter stops if it hears it
- ☐ **C** The checker ignores robots that do not send
- ☐ **D** `forward` only works after a send

6. Driving along the corridor, the robot faces heading 90. Why does it steer from `y` and not `x`? [1 mark]

- ☐ **A** Facing along x, sideways drift shows up in y
- ☐ **B** x is always zero
- ☐ **C** `position()` only gives y
- ☐ **D** Heading 90 means facing up the mat

The task: one at a time

Wait for `corridor clear`, send `entering`, drive through to the green zone without touching the walls or the Porter.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()
# radio this to every other robot
send('entering')
# drive forward at 60 for 72 cm, then stop
forward(60, distance=72)
# radio this to every other robot
send('out')
```

Plan your program here, then type it in and press Run.

Do it on the robot

www.bugbotlab.com/learn/9-5-taking-turns/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Print how long you waited.
2. Send `out` when you reach the zone, then come back through the corridor the same way.
3. If `busy` arrives while you are still waiting, print `missed it` and keep waiting for the next `clear`.